

Sistemet e Bazës së të Dhënave

Hyrje në SQL

Ramiz HOXHA

ramiz.hoxha@ubt-uni.net

2020/2021

FAKULTETI: SHKENCAVE KOMPJUTERIKE DHE INXHINIERIS

Qellimi i ligjërates

□ Intuita për SQL (gjysma e parë e sotme)

- Modeli themelor relativ (aka tabela)
- Konceptet SQL që do të studiojmë

□ Faza II: Konceptet bazik në SQL

- Skemat, struktura e pyetjes së SELECT-FROM-WHERE, etj
- Shembull SQL (eksplorimi i grupeve të të dhënave të vërteta)

Konceptet e modelit të Relacional

Entitetet (p.sh., Studentet, Lëndet, Profesoret)

Relacionet (p.sh., Kroi është i regjistruar në BDH)



Baza e të Dhenave Relacionale

❑ Baza e të dhënave është bashkësi e relacioneve.

❑ Relation= është *term* matematik për **tablë**.

- Tabela e përbër nga *kolona* dhe *rreshta*
- Relacioni përbëhet nga dy pjes:
 - Skema Relacionale
 - Instanca Relacionale

❑ **Skema e Relacionale:**

- Specifikon: *emrin e relacionit*, *emrin* dhe *tipin* e seciles *kolonë* (atributet).
- p.sh entiteti *Studenti*:

```
STUDENTI (ID_Studenti:char, emriStudentit:varchar, mbiermiStudentit:varchar,  
          dataLindjes:date, email:varchar, qyteti:varchar, drejtimi:char)
```

Instanca Relacionale

□ **Instanca e Relacionale**: një tabelë e përbërë nga *atribute* ose *kolona*

- Kolona = atributet = fusha
- Rresht = tuple.

Skema e Relacionit

STUDENTI (ID_Studenti:char, emriStudentit:varchar, mbiermiStudentit:varchar, dataLindjes:date, email:varchar, notaMesatare:float qyteti:varchar)

Instancat e Relacionit

atributet ose kolonat

ID_Studenti	emriStudentit	mbiermiStudentit	dataLindjes	email	notaMesatare	qyteti
S001	Agnesa	Begiraj	21.10.2001	ab@gmail.com	3.70	Peje
S002	Kroi	Sopa	16.09.2000	ks@hotmail.com	5.00	Prishtine
S003	Agimi	Keka	01.11.2000	ak@gmail.com	4.20	Gjilan
S004	Florentina	Buzuku	12.05.2001	fb@hotmail.com	4.30	Prishtine
S005	Bekim	Dema	10.03.2000	bd@yahoo.com	3.90	Gjakove
S006	Flora	Krasniqi	11.07.2001	fk@yahoo.com	4.50	Peje
S007	Kaltrina	Dushi	12.07.2000	kd@gmail.com	4.80	Gjakove
S008	Dardan	Daka	09.06.2001	dd@gmail.com	3.40	Prishtine
S009	Shkemb	Bajra	10.06.2000	sb@hotmail.com	3.20	Gjilan
S010	Enka	Kamza	08.09.2001	ek@hotmail.com	4.70	Prizeren

Tuple ose rreshtat

KARDINALITETI = 10

Hyrje në SQL

Hyrje në SQL

- ❑ SQL është një gjuhë standarte për kërkimin dhe manipulimin e të dhënave
 - SQL është një gjuhë programimit e nivelit të lartë
 - Gjuha origjinale ishte “**Structured English Query Language**” prononcohet: “**S.Q.L**” - (SEQUEL)
 - Gjuha **SEQUEL** është zhvilluar, si pjesë e projektit *Sistem R* në laboratorin për hulumtime San Jose të (fillim të 1970’ave) nga IBM.

**SQL qëndron për
Structured
Query
Language**

- ❑ Ka standarte të shumë:
 - ANSI SQL, SQL92 (SQL2), SQL99 (SQL3),

- ❑ Mbështetur nga të gjitha sistemet e mëdha të bazës së të dhënave komerciale

- ❑ **Vendoret** të RDBMS/SMBDDH

ORACLE®
DATABASE

Microsoft®
SQL Server

IBM DB2

MySQL®

PostgreSQL

Hyrje në SQL (2)

- ❑ Structured Query Language - SQL (Gjuha e *pyetjeve/kërkimi të strukturuar*)
- ❑ Mundëson përdoruesit të *komunikojnë, manipuloj* me të *dhënat* nga tabelat e Bazës së të Dhënave Relacionale



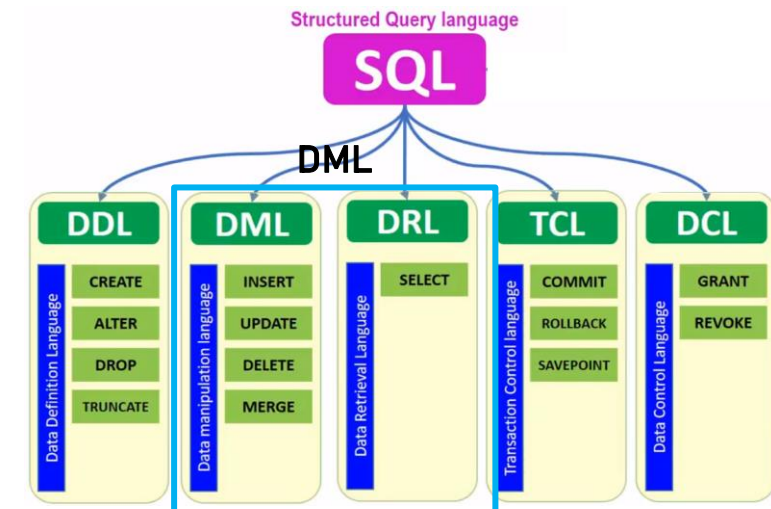
Hyrje në SQL...

□ Data Manipulation Language (DML)-Gjuhë për Manipulimin e të Dhënave

- Query një apo më shumë tavolina
- INSERT/DELET/UPDATE/SELECT tuples në tabela

□ Data Definition Language (DDL)- Gjuhë për Definimin e të Dhënave

- Përcaktoni skemat relacionale
- CRATE /ALTER/DROP tabelat dhe atributet e tyre



Relacioni = Tabela në SQL

- Tabela ne SQL Një **relacion** ose **tabelë** është një shumë e grupimit të rreshtave (tuples) që kanë **atributet** (kolonat) e **specifikuara** nga **skema**

Një **atribut** (ose **kolonë**) është një tip i të dhënave të pranishëm në secilën **tuple** në **relacion**

Shënim Atributet (ose kolonë) duhet të kenë një **lloj atomik** në SQL dmth. jo një listë, grup, etj.

ID_Studenti	emriStudentit	mbiermiStudentit	dataLindjes	email	notaMesatare	qyteti	drejtimi
S001	Agnesa	Beqiraj	21.10.2001	ab@gmail.com	3.70	Peje	MBE
S002	Kroi	Sopa	16.09.2000	ks@hotmail.com	5.00	Prishtine	SHKI
S003	Agimi	Keka	01.11.2000	ak@gmail.com	4.20	Gjilan	SHKI
S004	Florentina	Buzuku	12.05.2001	fb@hotmail.com	4.30	Prishtine	MEK
S005	Bekim	Dema	10.03.2000	bd@yahoo.com	3.90	Giakove	SI
S006	Flora	Krasniqi	11.07.2001	fk@yahoo.com	4.50	Peje	SHKI
S007	Kaltrina	Dushi	12.07.2000	kd@gmail.com	4.80	Giakove	MBE
S008	Dardan	Daka	09.06.2001	dd@gmail.com	3.40	Prishtine	SI
S009	Shkemb	Bajra	10.06.2000	sb@hotmail.com	3.20	Gjilan	MEK
S010	Enka	Kamza	08.09.2001	ek@hotmail.com	4.70	Prizeren	SHKI

Një **tuple** ose **rresht** ose **rekord** është një **input** e **vetme** në **tabelë** që ka atributet e specifikuara nga **skemat**

Data Types in SQL

□ Llojet atomike:

Karakteret: CHAR (20), VARCHAR (50)

Numrat: INT, BIGINT, SMALLINT, FLOAT

Të tjerët: MONEY, DATETIME

□ CHAR është 'varg/string' i karaktereve me **gjatësi të fikse** të tipit e të dhënave

- Tipi **char** për *çdo hapësirë të mbetur* në fushë është e alokuar me hapsir **boshe**.
- char** merr **1 bajt për karakter**. p.sh nëse kemi **char(10)** merr **10 bajt në disk**.

□ VARCHAR është një lloj i të dhënave **të vargut** me **gjatësi variabile**

- kështu që **varchar** mban vetëm karakteret që i caktojm për të.
- varchar** merr **1 byte për karakter**, + **2 byte** për të **mbajtur** gjatësin e informacionit.
- p.sh**, nëse vendosni një të dhën **varchar(10) = 'Agim'**, atëherë do të duhen **4 bajt** (për **A, g, i dhe m**) **plus 2 bajt** ose **6 bajt në total**.

□ Çdo **atribut** duhet të ketë një lloj atomik

Prandaj tavolina janë të flat (e rrafshët)

Skemat e Tabelave

Skema e një **tabele** është **emri** i *tabelës*, **atributet** e tij dhe **llojet** e tyre:

```
STUDENTI(ID_Studenti:char, emriStudentit:varchar, mbiermiStudentit:varchar,  
          dataLindjes:date, email:varchar, notaMesatare:float qyteti:varchar)
```

Një **çelës** është **një atribut** *vlerat* e të cilit janë **unike**; ne *nënvizojmë* një **çelës**

```
STUDENTI(ID_Studenti:char, emriStudentit:varchar, mbiermiStudentit:varchar,  
          dataLindjes:date, email:varchar, notaMesatare:float qyteti:varchar)
```

Çelësi: Kufizimet

Një **çelës** është një nëngrup **minimal i attributeve** që vepron si një **identifikues unik** për tuples në një relacion

- nëse dy rreshta bien dakord për vlerat e çelësit, atëherë ata duhet të jenë të njëjta **tuple** (rresht)!

```
STUDENTI(ID_Studenti:char, emriStudentit:varchar, mbiermiStudentit:varchar,  
dataLindjes:date, email:varchar, notaMesatare:float qyteti:varchar)
```

Zgjedhjet e dizajnit?

1. Cilin do të zgjidhnit si çelës?
2. A garantohet se ekziston gjithmonë një çelës?
3. A mund të kemi më shumë se një çelës?

Vlerat NULL dhe NOT NULL

- Për të thënë "nuk e di vlerën" ne përdorim **NULL**
 - **NULL** ka semantikë (ndonjëherë të dhimbshme), më shumë detaje më vonë.
- Në SQL, ne mund të detyrojmë që një kolonë të mos jetë NULL,
 - p.sh., "emriStudentit" në këtë tabelë

Tabela 7.2 - SQL Komandat e Manipulimit të të Dhënave (1)

Tabela 7.2 - SQL Komandat e manipulimit të të Dhënave	
KOMANDA ose OPSIONI	PËRSHKRIMI
INSERT	Inserton/shton rresht (a) në një tabelë
SELECT	Zgjedh atributet nga rreshta të një apo më shumë tabelave apo view
WHERE	Përcakton kufizimet-zgjedhjen e rreshtave në bazë të një shprehje të kushtëzuar
GROUP BY	Grupon rreshtat e përzgjedhura në bazë të një ose më shumë attributeve
HAVING	Kufizon zgjedhjen e rreshtave të grupuara në bazë të një kusht
ORDER BY	Reshton rreshtat e përzgjedhura në bazë të një ose më shumë attribute
UPDATE	Modifikon vlerat e attributeve të rreshtavenë të një ose më shumë tablea
DELETE	Fshin një apo më tepër rreshta nga një tabelë
COMMIT	Në mënyrë të përhershme ruan ndryshimet e të dhënave
ROLLBACK	Rikthen të dhënat në vlerat e tyre origjinale
OPERATORËT KRAHASUES	
=, <, >, <=, >=, <>	Përdoren në shprehjet e kushtëzuara
OPERATORËT LOGJIK	
AND/OR/NOT	Përdoren në shprehjet e kushtëzuara

Tabela 7.2 - SQL Komandat e manipulimit të të Dhënave (2)

OPERATORËT SPECIAL	Përdoren në shprehjet e kushtëzuara
BETWEEN	Kontrollon nëse vlera atributi është brenda një rangu
IS NULL	Kontrollon nëse vlera atributi është NULL
LIKE	Kontrollon nëse vlera e atributit përputhet me një moster të caktuar në varg
IN	Kontrollon nëse vlera atributi përputhet me ndonjë vlerë të brenda një listë të vlerave
EXISTS	Kontrollon nëse Subquery këthen ndonjë rresht
DISTINCT	Kufizon vlerat në vlera unike
FUNKSIONET AKUMULIMIT/AGGREGATE	Përdoret me SELECT për t'u kthyer përmbledhjet matematikore në kolona
COUNT	Kthen numrin e rreshtave me vlera JoNULL për një kolonë të caktuar
MIN	Kthen vlerën minimale të attributeve të gjetur në kolonë të caktuar
MAX	Kthen vlerën maksimale të attributeve të gjetur në kolonë të caktuar
SUM	Kthen shuma e të gjitha vlerave të kolonës të caktuar
AVG	Kthen mesataren e të gjitha vlerave të kolonës të caktuar

Insertimi i të Dhënave në SQL

❑ Për të *futur/shtuar* të dhënat në një BDH, duhet të përdorim deklaratën 'INSERT'.

❑ Më poshtë është edhe forma e shkurt:

- `INSERT INTO <emri_Tabelës>`
`VALUES (<vlëra_kolonës1>,<vlëra_kolonës2>,<vlëra_kolonës3>,...);`

❑ Emri i kolonës (p.sh.: *kolona1*, etj.) duhet të *korrespondoj* në (p.sh.: *vlërën_kolonës1*, etj).

- `INSERT INTO <emri_Tabelës> (<kolona1>,<kolona2>,<kolona3>,...)`
`VALUES (<vlëra_kolonës1>,<vlëra_kolonës2>,<vlëra_kolonës3>,...);`

❑ Një shembull tipik i *futjes së rekordeve* të *Urdhresës* që kemi krijuar më herët do të ishte:

```
insert into STUDENTI values ('S001','Agnesa','Beqiraj','10/21/2001','ab@gmail.com',3.70,'Peje');
```

```
insert into STUDENTI (emriStudentit,mbiermiStudentit,ID_Studenti,dataLindjes,  
email,notaMesatare,qyteti) values ('Kroi','Sopa','S002','09/16/2000','ks@hotmail.com',5.00,'Prishtine');
```

Deklarata SELECT

Deklarimi i Skemës në SQL

❑ Skema Studenti

```
STUDENTI(ID_Studenti:char, emriStudentit:varchar, mbiermiStudentit:varchar,  
dataLindjes:date, email:varchar, notaMesatare:float qyteti:varchar)
```

Create Table STUDENTI

```
(  
    ID_Studenti char(8) PRIMARY KEY,  
    emriStudentit varchar(30) NOT NULL,  
    mbiermiStudentit varchar(30) NOT NULL,  
    dataLindjes date, /*mm/dd/yyyy*/  
    email varchar(100),  
    notaMesatare float,  
    qyteti varchar(20)  
)
```

STUDENTI

	Column Name	Condensed Type	Nullable
🔑	ID_Studenti	char(8)	No
	emriStudentit	varchar(30)	No
	mbiermiStud...	varchar(30)	No
	dataLindjes	date	Yes
	email	varchar(100)	Yes
	notaMesatare	float	Yes
	qyteti	varchar(20)	Yes

Rasti i studimit

ID_Studenti	emriStudentit	mbiermiStudentit	dataLindjes	email	notaMesatare	qyteti
S001	Agnesa	Beqiraj	21/10/2001	ab@gmail.com	3.7	Peje
S002	Kroi	Sopa	16/09/2000	ks@hotmail.com	5	Prishtine
S003	Agim	Keka	01/11/2000	ak@gmail.com	4.2	Gjilan
S004	Florentina	Buzuku	12/05/2001	gb@hotmail.com	4.3	Prishtine
S005	Bekim	Dema	10/03/2000	bd@yahoo.com	3.9	Gjakove
S006	Flora	Krasniqi	11/07/2000	fl@yahoo.com	4.5	Peje
S007	Kaltrina	Dushi	12/07/2000	kd@gmail.com	4.8	Gjakove
S008	Dardan	Daka	07/06/2001	NULL	NULL	Prishtine
S009	Shkemb	Bajra	10/06/2000	sb@hotmail	3.2	Gjilan
S010	Enka	Kamza	09/08/2000	ek@hotmail.com	4.7	Prizeren

```
create database L02G1
use L02G1
```

```
create table Studenti
(
    ID_S char (5) Primary Key,
    emri varchar(40) NOT NULL,
    mbemri varchar (45) NOT NULL,
    dataL date,
    Email varchar (50),
    notaM float,
    qyteti varchar(20)
);
```

```
insert into Studenti values ('S002', 'Kroi', 'Sopa', '09/16/2000', 'ks@hotmail.com', 5.00, 'Prishtine')
insert into Studenti values ('S003', 'Agimi', 'Keka', '11/01/2000', 'ak@gmail.com', 4.20, 'Gjilan')
insert into Studenti values ('S004', 'Florentina', 'Buzuku', '05/12/2001', 'fb@hotmail.com', 4.30, 'Prishtine')
insert into Studenti values ('S005', 'Bekim', 'Dema', '03/10/2000', 'bd@yahoo.com', 3.90, 'Gjakove')
insert into Studenti values ('S006', 'Flora', 'Krasniqi', '07/11/2001', 'fk@yahoo.com', 4.50, 'Peje')
insert into Studenti values ('S007', 'Kaltrina', 'Dushi', '07/12/2000', 'kd@gmail.com', 4.80, 'Gjakove')
insert into Studenti values ('S008', 'Dardan', 'Daka', '07/06/2001', ' ', ' ', 'Prishtine')
insert into Studenti values ('S009', 'Shkemb', 'Bajra', '10/06/2000', 'sb@hotmail', '3.2', 'Gjilan')
insert into Studenti values ('S010', 'Enka', 'Kamza', '09/08/2000', 'ek@hotmail.com', '4.7', 'Prizeren')
```

Deklarata SELECT në SQL

□ Baza e Deklarates SELECT

- ③ **SELECT** $a_1, a_2, a_3, \dots, a_n$ ← Çfarë të kthehet (return) apo shfaqet në rezultat
- ① **FROM** $R_1, R_2, R_3, \dots, R_n$ ← Nga cilet Relacione apo Tabela
- ② **WHERE** kushti ← Kombinim të filtrimeve apo kushteve

□ Në Algjeber Relacionale:

$$\prod_{\{a_1, a_2, \dots, a_n\}} (\sigma_{\{kushti\}} (R_1 \times R_2 \times \dots \times R_n))$$

Deklarata SELECT në SQL...

□ Klausola Select duhët të përmas min një SELECT dhe FROM

- Rasti i perdorimi të klausoles SELECT dhe '*'.

/*Definimi i Klausoles SELECT: Shfaqjen e te gjitha te dhenave */

SELECT *

FROM STUDENTI

Results		Messages					
	ID_Studenti	emriStudentit	mbiermiStudentit	dataLindjes	email	notaMesatare	qyteti
1	S001	Agnesa	Beqiraj	2001-10-21	ab@gmail.com	3.7	Peje
2	S002	Kroi	Sopa	2000-09-16	ks@hotmail.com	5	Prishtine
3	S003	Agim	Keka	2000-11-01	ak@gmail.com	4.2	Gjilan

Formatimi i date ne MS SQL Server

*/*Konvertimi ne formatin europian 6*/*

```
Select emriStudentit, mbiermiStudentit, CONVERT(varchar,dataLindjes,6) as [DL]  
From STUDENTI  
where dataLindjes = '21.Oct.2001'
```

	emriStudentit	mbiermiStudentit	DL
1	Aghesa	Beqiraj	21 Oct 01

	emriStudentit	mbiermiStudentit	DL
1	Aghesa	Beqiraj	21 Oct 01
2	Kroi	Sopa	16 Sep 00
3	Agim	Keka	01 Nov 00
4	Florentina	Buzuku	12 May 01
5	Bekim	Dema	10 Mar 00
6	Flora	Krasniqi	11 Jul 00
7	Kaltrina	Dushi	12 Jul 00
8	Dardan	Daka	07 Jun 01
9	Shkemb	Bajra	10 Jun 00
10	Enka	Kamza	09 Aug 00

*/*Konvertimi ne formatin europian nr 13*/*

```
Select emriStudentit, mbiermiStudentit, CONVERT(varchar,dataLindjes,13) as [DL]  
From STUDENTI  
where dataLindjes = '21.Oct.2001'
```

	emriStudentit	mbiermiStudentit	DL
1	Aghesa	Beqiraj	21 Oct 2001

	emriStudentit	mbiermiStudentit	DL
1	Aghesa	Beqiraj	21 Oct 2001
2	Kroi	Sopa	16 Sep 2000
3	Agim	Keka	01 Nov 2000
4	Florentina	Buzuku	12 May 2001
5	Bekim	Dema	10 Mar 2000
6	Flora	Krasniqi	11 Jul 2000
7	Kaltrina	Dushi	12 Jul 2000
8	Dardan	Daka	07 Jun 2001
9	Shkemb	Bajra	10 Jun 2000
10	Enka	Kamza	09 Aug 2000

Deklarata SELECT DISTINCT në SQL

- Deklarata **SELECT DISTINCT** përdoret për të kthyer *vetëm vlera të veçanta* (të ndryshme)

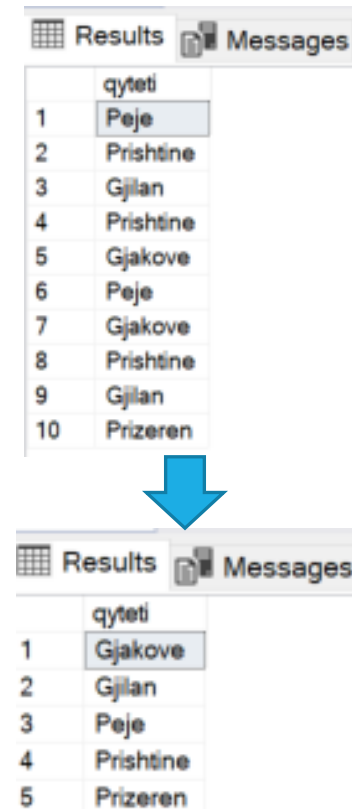
Shembulli i klauzoles SELECT pa DISTINCT. Deklarata e mëposhtme SQL zgjedh të gjitha (përfshirë vlerat duplikuara) nga kolona "Qyteti" në tabelën "STUDENTI":

```
Select S.qyteti  
From Studenti S
```

/*Shembulli i përdorimit të Deklarates SELECT DISTINCT

Rasti i përdorimit **DISTINCT**, për zgjedh ose selekton **vlerat veçanta** nga kolona **Qyteti** ne tabelen **STUDENTI** */

```
Select DISTINCT S.qyteti  
From Studenti S
```



The diagram illustrates the effect of the DISTINCT keyword. It shows two screenshots of a database query result. The top screenshot shows a table with 10 rows of city names, including duplicates. A blue arrow points down to the bottom screenshot, which shows the same table but with only 5 rows, where all duplicate values have been removed, leaving only the unique city names.

	qyteti
1	Peje
2	Prishtine
3	Gjilan
4	Prishtine
5	Gjakove
6	Peje
7	Gjakove
8	Prishtine
9	Gjilan
10	Prizeren

	qyteti
1	Gjakove
2	Gjilan
3	Peje
4	Prishtine
5	Prizeren

SQL-DML: Klauzola **WHERE** (Operatorët SQL - krahasim)

❑ Operatorët SQL:

- Ka *dy lloj operatorëve*, përkatësisht **operatorët krahasues** dhe **operatorët logjikë**. Këta operatorë janë përdorur kryesisht në, **klauzolën WHERE**, dhe **klauzolën HAVING** për të filtruar të dhënat sipas perzgjedhjes të përcaktuar nga kushtet.

❑ Operatorët krahasim:

- **Operatorët e krahasimit** janë përdorur për të krahasuar të dhënat në kolona me vlera të veçanta në filtrime të caktuara,

Operatorët e krahasimit	Përshkrimi
=	barabartë me
<>, !=	nuk është e barabartë me
<	më i vogël se/ me pak se
>	më i madh se
>=	më i madh se ose e barabartë me
<=	me pak se ose e barabartë me

SQL-DML: Klauzola **WHERE**(Operatorët SQL - logjikë)(1)

□ Operatorët logjikë

- Janë tre **Operatorët Logjik** dhe *respektivisht* **AND**, **OR**, dhe **NOT**.
- Këta operatorë mundsojnë krahasimin e dy kushteve njëkohësisht për të përcaktuar output të përzgjedhur për një rresht.
- Kur nxjerrim të dhëna duke përdorur deklaratën **SELECT**, ju mund të përdorni operatorët logjike në **WHERE**, e cila ju mundson për të kombinuar më shumë se një kusht

Operatorët logjik	Përshkrim
OR	Për rreshtin e përzgjidhet të paktën një prej kushteve duhet të jetë e saktë (true).
AND	Për rreshtin e përzgjidhet të gjitha një kushtet duhet të jenë të saktë (true).
NOT	Për rreshtin e përzgjidhet kushtet e specifikuar duhet të jenë të false

SQL-DML: Klauzola WHERE

❑ Klauzola **WHERE** është përdorur në *deklarata* **UPDATE**, **DELETE**, dhe **SELECT**

Klauzola **WHERE** është përdorur kur ju dëshironi të nxjerrni *informacione specifike nga një tabelë*, përjashtim të dhënave të tjera irelevante. *p.sh*, kur ju doni të *nxjerrim informacion në lidhje me personat* me vend lindje në Prishtinë 'PR'.

/*Rasti i përdorimit të Klauzoles WHERE:

p.sh: Të shfaqeni Emri, Mbiemri dhe Qytetin për student me *note mesatare* me të mëdhe se 4.*/

```
Select S.emriStudentit, S.mbiemriStudentit, S.qyteti, S.notaMesatare  
From Studenti S  
Where s.notaMesatare >= 4;
```

	emriStudentit	mbiemriStudentit	qyteti	notaMesatare
1	Agnesa	Beqiraj	Peje	3.7
2	Kroi	Sopa	Prishtine	5
3	Agim	Keka	Gjilan	4.2
4	Florentina	Buzuku	Prishtine	4.3
5	Bekim	Dema	Gjakove	3.9
6	Flora	Krasniqi	Peje	4.5
7	Kaltrina	Dushi	Gjakove	NULL
8	Dardan	Daka	Prishtine	NULL
9	Shkemb	Bajra	Gjilan	3.2
10	Enka	Kamza	Prizeren	4.7

Pas përdorimit të
filtrimit sipas kushtin
nota mesatare > 4

	emriStudentit	mbiemriStudentit	qyteti	notaMesatare
1	Kroi	Sopa	Prishtine	5
2	Agim	Keka	Gjilan	4.2
3	Florentina	Buzuku	Prishtine	4.3
4	Flora	Krasniqi	Peje	4.5
5	Enka	Kamza	Prizeren	4.7

SQL WHERE (Operatorët SQL – logjikë AND dhe OR)

<atributi> operatori_krahasimit <vlera1> AND <atributi> operatori_krahasimit <vlera2>

/*Te shaqeni te dhenat si emri mbiemri qyteti dhe nota mesatare per student qe kan note me te madhe se 4 dhe jane nga qyteti Prishtine*/

Select s.emriStudentit, s.mbiemriStudentit, s.notaMesatare as [Nota Mesatare], s.qyteti as [Qyteti]

From Studenti S

where s.notaMesatare > 4 and s.qyteti = 'Prishtine'

	emriStudentit	mbiemriStudentit	notaMesatare	qyteti
1	Kroi	Sopa	5	Prishtine
2	Florentina	Buzuku	4.3	Prishtine

<atributi> operatori_krahasimit <vlera1> OR <atributi> operatori_krahasimit <vlera2>

/*Te shaqeni te dhenat si emri, mbiemri, qyteti dhe nota mesatare per student qe kan note me te madhe se 4.5 ose jane nga qyteti Gjilanit*/

Select s.emriStudentit, s.mbiemriStudentit, s.notaMesatare, s.qyteti as [Qyteti]

From Studenti S

where s.notaMesatare > 4.5 OR s.qyteti = 'Gjilan'

order by [NM]

	emriStudentit	mbiemriStudentit	notaMesatare	Qyteti
1	Shkemb	Bajra	3.2	Gjilan
2	Agim	Keka	4.2	Gjilan
3	Enka	Kamza	4.7	Prizeren
4	Kaltrina	Dushi	4.8	Gjakove
5	Kroi	Sopa	5	Prishtine

SQL WHERE (Operatorët SQL – logjikë NOT)

❑ **NOT** <atributi> operatori_krahasimit <vlera>

/*Te i shfaqeni te dhenat (emri , mbiemri) e studenteve qe nuk jane nga Gjilani*/

```
Select emriStudentit, mbiermiStudentit, qyteti  
from Studenti  
where NOT qyteti='Gjilan'
```

	emriStudentit	mbiermiStudentit	qyteti
1	Agnesa	Beqiraj	Peje
2	Kroi	Sopa	Prishtine
3	Agim	Keka	Gjilan
4	Florentina	Buzuku	Prishtine
5	Bekim	Dema	Gjakove
6	Flora	Krasniqi	Peje
7	Kaltrina	Dushi	Gjakove
8	Dardan	Daka	Prishtine
9	Shkemb	Bajra	Gjilan
10	Enka	Kamza	Prizeren

	emriStudentit	mbiermiStudentit	qyteti
1	Agnesa	Beqiraj	Peje
2	Kroi	Sopa	Prishtine
3	Florentina	Buzuku	Prishtine
4	Bekim	Dema	Gjakove
5	Flora	Krasniqi	Peje
6	Kaltrina	Dushi	Gjakove
7	Dardan	Daka	Prishtine
8	Enka	Kamza	Prizeren

/*Te i shfaqeni te dhenat (emri , mbiemri, nota mesatare) e studenteve qe kan not mesatare me te madhe se 4 por nuk jane te Prishtines*/

```
Select emriStudentit, mbiermiStudentit, qyteti, notaMesatare  
from Studenti  
where notaMesatare>4 AND NOT qyteti='Prishtine'
```

	emriStudentit	mbiermiStudentit	qyteti	notaMesatare
1	Kroi	Sopa	Prishtine	5
2	Agim	Keka	Gjilan	4.2
3	Florentina	Buzuku	Prishtine	4.3
4	Flora	Krasniqi	Peje	4.5
5	Kaltrina	Dushi	Gjakove	4.8
6	Enka	Kamza	Prizeren	4.7

	emriStudentit	mbiermiStudentit	qyteti	notaMesatare
1	Agim	Keka	Gjilan	4.2
2	Flora	Krasniqi	Peje	4.5
3	Kaltrina	Dushi	Gjakove	4.8
4	Enka	Kamza	Prizeren	4.7

SQL: Klauzola **WHERE** (Operatorët special)

□ Operatorët special në SQL

- Ekzistojnë fjalë kyçe ose operatorët special për krahasim në SQL të cilat përdoren për të përmirësuar aftësitë e kërkimit të një query SQL, ato janë:
- **"LIKE", "IN", "BETWEEN ... AND", "IS NULL"**.

Operatorët e special	Përshkrimi
LIKE	Vlera e kolonës është e njëjtë me karakterin të specifikuar (a).
IN	Vlera e kolonës është e barabartë me njerin nga vlerat të specifikuar të grupit.
BETWEEN...AND	Vlera e kolonës është në mes të dy vlerave, duke përfshirë vlerat fundit të specifikuar në varg.
IS NULL	Vlera në kolon nuk ekziston.

SQL: Klauzola **WHERE** (Operatorët special – **LIKE**)

❑ Operatori **LIKE** - përdoret në të krahasuar përputhshmerin në nën-vargje të attributeve të tipit **CHAR** ose **VARCHAR**.

❑ Sintaksa e operatorit **LIKE**: **<atributi> LIKE < përputhshmeri vargut>**

- *Nënvizimi* (_) përputhet me një karakter të vetëm
- *Përqindja* (%) përputhet me një sekuencë të karaktereve (përfshirë edhe zeron)
- Për (_) ose % është i nevojshëm si karakter alfabetik në vargun e përputhshmëris, përdorni para tyre karakterin (\).

/*Te i shfaqeni te dhenat e studentit ku qyteti fillon me shkronjeh P */

```
Select emriStudentit, mbiermiStudentit, qyteti  
from Studenti  
where qyteti LIKE 'P%'
```

	emriStudentit	mbiermiStudentit	qyteti
1	Aghesa	Beqiraj	Peje
2	Kroi	Sopa	Prishtine
3	Florentina	Buzuku	Prishtine
4	Flora	Krasniqi	Peje
5	Dardan	Daka	Prishtine
6	Enka	Kamza	Prizeren

/*Te i shfaqeni te dhenat e studentit ku shkronja e pare nuk i dihet */

```
Select emriStudentit, mbiermiStudentit, qyteti  
from Studenti  
where mbiermiStudentit LIKE '_opa'
```

	emriStudentit	mbiermiStudentit	qyteti
1	Kroi	Sopa	Prishtine

SQL: Klauzola **WHERE** (Operatorët special – **BETWEEN**)

□ Sintaksa: <atributi> BETWEEN <vlera1> AND <vlera2>

/*Te i shfaqeni studentet me note mesatare nga 4 deri 4.7*/

```
Select emriStudentit, mbiermiStudentit, notaMesatare  
from Studenti  
where notaMesatare BETWEEN 4 AND 4.7
```

-- BETWEEN përfshin dy skajet e prerje p.sh rastin tone vlern 4 dhe 4.7

	emriStudentit	mbiermiStudentit	notaMesatare
1	Agim	Keka	4.2
2	Florentina	Buzuku	4.3
3	Flora	Krasniqi	4.5
4	Enka	Kamza	4.7

SQL: Klauzola **WHERE** (Operatorët special - **IN**)

□ Sintaksa e operatorit **IN**:

■ <atributi> **IN** <'vlera1', 'vlera2'... 'vleratn'>

/*Te i shfaqeni studentet me note mesatare nga 4.2, 4.5 dhe 5*/

```
Select emriStudentit, mbiermiStudentit, notaMesatare  
from Studenti  
where notaMesatare IN('4.2', '4.5', '5')
```

	emriStudentit	mbiermiStudentit	notaMesatare
1	Kroi	Sopa	5
2	Agim	Keka	4.2
3	Flora	Krasniqi	4.5

SQL: Deklarata UPDATE (1)

- Deklarata për *përditësim* 'UPDATE' është përdorur për të *modifikuar* të dhënat në një tabelë. Sintaksa e përgjithshme është:

```
UPDATE      tablename
SET         columnname = expression [, columnname = expression]
[WHERE      conditionlist];
```

```
UPDATE <emir-tabelës>
SET <kolona1> = <vlera1>, <kolona2> = <vlera2>, ...
WHERE <kriteri>
```

SQL: Deklarata UPDATE...

- Deklarata për *përditësim* 'UPDATE' është përdorur për të *modifikuar* të dhënat në një tabelë. Sintaksa e përgjithshme është:

Përditson titullin e CDs nga 'Demokraci' në 'Demokraci-ROCK'

```
UPDATE CD
```

```
SET titulli='Demokraci-ROCK'
```

```
WHERE cd_ID=10;
```

cd_ID	titulli	kohezgjatja	qmimi	viti
10	Demokraci	01:32:00.0000000	7.20	2016
20	Bajrush doda	03:50:00.0000000	12.00	1989
30	Fenix	00:05:00.0000000	2.00	2013
40	Vibe	00:04:20.0000000	3.69	2010
50	Stupcat Akademiku	00:45:25.0000000	8.99	2014

cd_ID	titulli	kohezgjatja	qmimi	viti
10	Demokraci-ROCK	01:32:00.0000000	7.20	2016
20	Bajrush doda	03:50:00.0000000	12.00	1989
30	Fenix	00:05:00.0000000	2.00	2013
40	Vibe	00:04:20.0000000	3.69	2010
50	Stupcat Akademiku	00:45:25.0000000	8.99	2014

Deklarata DELETE

□ Deklarata DELETE

- 'DELETE' është përdorur të **fshijë** elemente nga bazën e të dhënave. Sintaksa është shumë e ngjashme me **update** dhe **select** deklaratat:

```
DELETE FROM tablename  
[WHERE conditionlist ];
```

```
DELETE FROM <emir-tabelës>  
WHERE < kriteri >
```

- Në thelb ne kemi zgjedhur cilat të dhënat ne duam të **fshijëm** duke përdorur **WHERE** klauzolën. Shembull i një rasti të përdorimit do të jetë:

```
DELETE FROM PERSON  
WHERE PERSONID=12345;
```

❑ Operatorët aritmetikë

- Si në shembullin, ju mund të përdorni operatorët aritmetikë me atributet tabelë në listën kolona ose në një shprehje të kushtëzuar. Në fakt, komandat SQL janë përdorur shpesh në lidhje me operatorët aritmetikë. Sintaksa :

```
SELECT <Expression>[arithmetic operator]<expression>...
```

```
FROM [table_name]
```

```
WHERE [expression];
```

```
--Shfaqe çmimin e CDs se 'Bajrush doda' pas aplikimit te  
zbitjes 5 euro
```

```
Select titulli, (qmimi - 5.00) as 'qmimi i CD pas
```

```
Zbritjes per 5 euro'
```

```
From CD
```

```
Where titulli = 'Bajrush doda'
```

```
Select titulli, (qmimi - 5.00) as 'qmimi i CD pas Zbritjes  
per 5 euro'
```

```
From CD
```

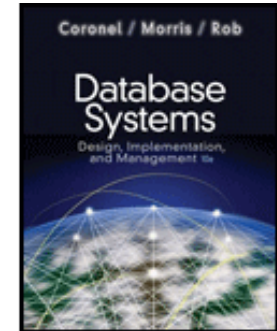
```
Where qmimi =7+5
```

Operatorët aritmetikë	Përshkrim
+	Mblidhni (Add)
-	Zbriteni (Subtract)
*	Shumezoj (Multiply)
/	Pjesëtoj (Divide)
^	Ngritja e fuqisë së (disa aplikacionet përdorin ** vend të ^)
% (Modulo)	Kthen mbetjen numër të plotë , në operacionin e pjesëtoj. Për shembull, 17% 5 = 2, sepse pjesa tjetër nga 17 pjesët 5 është 3.

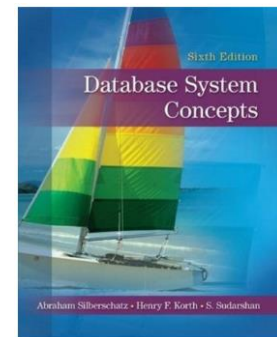
Results Messages	
titulli	qmimi i CD pas Zbritjes per 5 euro
Bajrush doda	7.00

Referencat

- Design, Implementation, and Management. 10th Edition (Chapter 7).



- *Database System Concepts* book (chapter 3)



Pyetje

